From Delaunay triangulation to adaptive anisotropic mesh generation

P.L. George

INRIA, Projet Gamma,
Domaine de Voluceau, Rocquencourt,
BP 105, 78153 Le Chesnay Cedex, France.
email: paul-louis.george@inria.fr,

Key words Delaunay triangulation, Mesh generation, Mesh adaptation, Adaptivity, h-method, Adaptive computation, Anisotropic meshes, FEM applications, CFD applications

Abstract *We are concerned with Delaunay mesh generation methods and mesh adaptivity issues. Planar, surface and volume meshing have been automated to a large extent. Over the last few years, meshing activities have been focused on adaptive schemes where the features of a solution field must be accurately captured. To this end, meshing techniques must be revisited in order to be capable of completing high quality meshes conforming to these features. Error estimates are therefore used to analyze the solution field at a given stage and, based on the results the infunction they yield, adapted meshes are created before computing the next stage of the solution field. A number of novel meshing issues must be addressed including how to construct a mesh adapted to what the error estimate prescribes. While other approaches exist, only Delaunay based meshing techniques are investigated in this paper.*

1 Introduction

To carry out a Finite Element Analysis (FEA or FEM), or any type of analysis such as a Boundary Element Method (BEM) or a Finite Volume Method (FVM) which requires using a spatial decomposition of the domain of interest, it is necessary to construct an appropriate mesh of the corresponding computational domain. This is the first step we face when using such methods. There are various mature automatic mesh generation methods which are widely used in software packages, [17]. Nevertheless, the demand in terms of meshing facilities is changing. At present,

developing quality meshes adapted to the physics of the problem to be solved represents a field of intensive research, [36].

Meshes can be categorized as being *structured* or *unstructured*. Structured meshes mainly concern geometries that are more or less close to quadrilateral (hexahedral) regions or decomposed by means of such simply shaped regions while arbitrarily shaped domains are more tedious to consider in this way. Therefore, unstructured meshes appear to be the solution. Construction methods in this case fall into three categories : (based on) *hierarchical spatial decompositions* (quadtree, octree), [43], (on) *advancing-front* strategies, [38], [27], [29], and (by means of) *Delaunay* type methods, [41], [30], [23] which is the method we have in favour.

Delaunay type methods make use of Delaunay triangulation algorithms. Such an algorithm results in the insertion of a point in a given mesh and completes a mesh of the convex hull of the set of points formed by the vertices of the domain boundary discretization (the input data). This material can be used as a point to point connector for mesh construction. Once the boundary points have been inserted, the boundary discretization can be obtained (a rather tedious task, at least in three dimensions), then points are created inside the domain before being connected in turn. The final mesh is obtained when the domain is saturated (in the sense it is no longer necessary to add new points).

The above discussion concerns the so-called *classical* mesh generation problem. Given a discretized boundary, we complete at best a mesh in the corresponding domain composed of elements that are judged to be reasonable in terms of *size* and *quality* (shape). The size is related to the greater or lesser thickness of the boundary discretization and remains to be properly defined inside the domain. The quality is related to the element aspect ratios. Moreover, the way in which these two parameters vary in a given region is an important issue. It could be observed that these criteria greatly depend on the way in which the field points are created, such an issue being somewhat tedious. Turning to mesh *adaptation* involves looking at the same problem while adding a series of constraints about element size and element directions (*anisotropic* case), [31], [4], [37], [32]. The aim is no longer to obtain the best possible mesh but to complete elements of a prescribed size and directions where necessary.

2 Delaunay based mesh generation methods

Delaunay based mesh generation methods were introduced by various authors, [25], [40], [26], [41] and [30].

2.1 Planar and volume domains

Before entering into the description, we give an indication of what we term a mesh generation problem. We are given a domain Ω in $\mathbb{R}^2$ or $\mathbb{R}^3$. This domain is defined by its boundary, Γ, which, in turn, is known by means of a discretization. The latter is a collection of line segments in two dimensions, these segments defining

a polygonal approximation of Γ. In three dimensions, the boundary is defined by means of a triangulated surface (a list of triangles or quadrilaterals) which, again, defines an approximation of Γ. The problem is then, using this sole data, to construct an appropriate mesh of Ω (actually, an approximation of Ω). Such a context defines a so-called *classical* mesh generation problem where the goal is to recover Ω by quality elements where the notion of quality only refers to aspect ratio (or element shape), mesh gradation but does not explicitly include any information about element sizing. Specifying sizing or directional requirements leads to a so-called *adapted* or *controlled* mesh generation problem where the targeted elements, as above, must be well-shaped, nicely graded and, in addition, must conform to a given sizing or directional specification (referred to as a metric in the following). The description below mainly concerns a classical mesh generation problem while a large part the following of the chapter discusses the other mesh generation issue (*i.e.,* how to complete an adapted mesh and therefore access adaptive computations)

2.2 Surface domains

Surface domains are different from planar or volume domains in the sense that a surface mesh must conform to the geometry of the surface in hand. In other words, smoothness, quality and gradation are demanded but, it is mandatory to match the geometry. This leads to paying particular attention to the surface curvatures thus making a purely two-dimensional approach unlikely to be suitable. In short, there are two ways to define a surface. A parametric definition involves defining a parametric space (thus in two dimensions) together with an appropriate mapping function. A discrete definition allows the geometry to be described by means of a mesh which is, *a priori*, a *geometric* mesh and not a *computational* mesh. To some extent, parametric surfaces can be meshed using a two-dimensional method provided information about the geometry is used to govern the method. On the other hand, discrete surfaces must be considered using the geometric properties of the surface directly.

2.3 Delaunay type methods

Many people find Delaunay type methods very appealing as the keyword Delaunay has a touch of elegance based on various theoretical issues, see [33] and [8] about properties of such triangulations. Despite this, a more subtle analysis and actual experience both indicate that those theoretical issues are unlikely to be directly usable in the context of creating FE meshes. On the other hand, it is frequent to observe that many people do not have a precise idea about the difference between a triangulation problem (related to a set of points) and a meshing problem (related to a domain). Nevertheless, Delaunay triangulation algorithms can be revisited so as to be included in the ingredients of the so-called Delaunay mesh generation methods.

Delaunay triangulation. While alternative methods exist, a popular and straightforward method for generating a Delaunay triangulation of a given set of points in $\mathbb{R}^2$ or $\mathbb{R}^3$ is the so-called Bowyer-Watson algorithm also developed by Hermeline at the same time (1981). It is an incremental method that results in inserting one point in a given Delaunay triangulation.

Let $\mathcal{T}$ be the Delaunay triangulation of the set of n points and let P be a point to be inserted, under some realistic assumptions, this method simply reads

$$\mathcal{T} = \mathcal{T} - \mathcal{C}(P) + \mathcal{B}(P)$$

where $\mathcal{C}(P)$ is the cavity associated with point P and $\mathcal{B}(P)$ is the re-meshing of $\mathcal{C}(P)$ based on P. Cavity $\mathcal{C}(P)$ is the set of simplices in T whose open circumdisk (circumball in three dimensions) contains point P while ball $\mathcal{B}(P)$ is simply the simplices constructed by connecting P with the boundary edges (or triangles) of $\mathcal{C}(P)$. Theoretical issues show that provided the former $\mathcal{T}$ is Delaunay, then the resulting $\mathcal{T}$ with P as a vertex is also Delaunay. This method allows planar and volume (or arbitrary dimensional) triangulation to be generated.

The key of the construction is the the proper definition of the cavity. It is constructed following a constrained proximity criterion given by :

$$\{K, K \in \mathcal{T}, P \in Ball(K) \quad \text{and} \quad P \text{ visible from each vertex in } K\}$$

where $Ball(K)$ is the opened circumball of K.

To characterize the cavity $\mathcal{C}(P)$, we introduce the *Delaunay measure* α associated with pair (P, K) :

$$\alpha(P, K) = \left[\frac{d(O_K, P)}{r_K}\right],$$

where O_K (resp. r_K) is the center (resp. radius) of the circumsphere of K and $[*]_{\mathcal{M}_d}$. The proximity criterion, $P \in Ball(K)$, is then expressed by $\alpha(P, K) < 1$. Actually, cavity $\mathcal{C}(P)$ is defined by $\mathcal{C}(P) = \mathcal{C}_1(P) \cup \mathcal{C}_2(P)$ with

$$\mathcal{C}_1(P) = \{K, K \in \mathcal{T}, K \text{ including } P\}$$

$$\mathcal{C}_2(P) = \{K, K \in \mathcal{T}, \exists K' \in \mathcal{C}(P), K \text{ adjacent to } K',$$

$$\alpha(P, K) < 1, \quad P \text{ visible by the vertices of } K\}.$$

Hence, region $\mathcal{C}(P)$ is completed by adjacency starting from the elements in $\mathcal{C}_1(P)$. After this definition, and in theory, the cavity is star-shaped with respect to P and $\mathcal{B}(P)$ is valid.

Additional constraints. Variations of this incremental algorithm include constrained versions where additional constraints are demanded which can be of a topological nature (e.g. specified edges (or triangle facets) are maintained through the process) or of a metric nature (e.g. additional properties such as element quality control are included in the construction).

To this end, the notion of adjacency is modified to account for the constraints when constructing the cavity.

Robustness issues. While the theory says that the above construction is well founded, round-off errors may lead to an unvalid triangulation. Therefore, and in specific in three dimensions and for anisotropic meshes, a correction, [20], of the cavity based on element removal can ensure the desired properties.

Following this, non Delaunay triangulations can be successfully carried out provided an adequate construction of $\mathcal{C}(P)$ and anisotropic cases can be addressed.

Delaunay based meshing algorithms. The appropriate above constrained incremental method can be used as part of a meshing algorithm (thus referred to as Delaunay-based). Let Ω be the domain to be meshed and let Γ be a discretization of its boundary. The set of points in this boundary discretization is triangulated using the above incremental method (after being included in a convex bounding box). The resulting triangulation is a triangulation of the introduced box where, in general, extracting a mesh of Ω is unlikely to be possible. This is due to the fact that the boundary entities are not necessarily edges or facets of this triangulation. In other words, inserting the two endpoints of a boundary edge (the three vertices of a boundary triangle) may result in a triangulation where this edge (triangle) does not exist.

Various methods have been developed to allow for the regeneration of such missing entities. In two dimensions, edge swapping operators result in what is needed while in three dimensions the same is rather tedious. Nevertheless, those methods readily work and result in a triangulation of the box where all the boundary entities are established. As a consequence, a mesh of Ω can be obtained by suppressing those elements in the box outside the domain.

At this time, we have in hand a mesh of Ω whose vertices are, roughly speaking, the boundary vertices. Such a mesh is unlikely to be suitable for FE computations. Therefore field points must be created before being inserted. Methods to create the necessary field points include using the centroid or the circumcenter of some mesh elements based on appropriate criteria (such as density requirement, element quality concern, etc). Other methods make use of a tree-structure to define those field points or introduce points along the edge elements (for any dimensions) or, again, use an advancing front strategy to locate those points. At completion, e.g. when the mesh is saturated (according to the selected criteria), the resulting mesh is optimized (with regard to quality measures) by means of edge or facet swapping, node repositioning, etc., as a Delaunay triangulation is not, in general and particularly in three dimensions, a quality triangulation (for FE purposes).

Combination of methods. The three main classes of automated mesh generation methods have both advantages and drawbacks. Combining one or more of these methods is therefore an elegant way to benefit from the advantages of such or such a method while avoiding any possible drawbacks. In this respect, Delaunay type methods have been combined with octree or advancing front techniques.

Surface meshing versus Delaunay type methods. The notion of a Delaunay surface mesh is in some ways confusing. The Delaunay criterion, the key to Delaunay triangulation methods, indicates that the circumdisk (circumball in three dimensions) of the mesh elements is empty. For a surface, this notion is meaningless for a number of reasons. First such disks are not defined, second, the Delaunay criterion (that correlates to a proximity and visibility criterion), does not include any concern about curvatures (and thus directions).

Nevertheless, parametric surfaces can be meshed by means of a Delaunay method where the Delaunay property applies in the parametric space (thus a planar region) but is necessarily an anisotropic version of the method (to handle the curvature of the true surface).

Also, while the notion of a Delaunay surface mesh is confusing, the notion of Delaunay conforming (or admissible) surface meshes is well founded. Such a mesh is such that inserting, by means of a three dimensional Delaunay algorithm, the endpoints of the given surface triangles results in a tetrahedral mesh where all the triangles in the given surface mesh are facets of tets.

3 Mesh quality and adaptivity

A quality mesh, as we will see, is a mesh adapted to given quality and density requirements. It is introduced as an occurence of a more general class of meshes. We first introduce the notion of a regular mesh and we show how this allows for the definition of an adapted mesh.

3.1 Regular mesh

Let Ω be a closed bounded domain in $\mathbb{R}^2$ or $\mathbb{R}^3$ defined by its boundary Γ. A quality simplicial mesh or a *regular* mesh of Ω is a mesh whose elements are equilateral (regular). The existence of such a mesh is not guaranteed in general. Indeed, it depends, to some degree, on the domain boundary discretization. Therefore, we will call a simplicial regular mesh the "best" simplicial mesh that can be completed. As the issue of constructing a regular mesh for an arbitrary domain is an open problem, there exist various methods which allow the construction of "almost" regular meshes.

In a classical context, two types of boundary discretizations can be envisaged. The first case concerns uniform discretizations where a constant step-size is given. The main advantage of such a discretization is that, in principle, it is possible to complete a regular mesh. Nevertheless, this does not guarantee a good approximation of the domain boundaries for a given step-size or may lead to meshes with an "infinite" number of elements (thus impossible to consider). Given a uniform discretization of a domain boundary, a regular mesh is nothing more than a mesh where the element sizes are "equal" to the step-size serving at the boundary discretization. Thus, the desired size for the elements in the mesh is known *a priori* at each mesh vertex in the regular domain mesh. Let us consider the case where the

domain boundary is composed of several connected components and where these are discretized by means of different stepsizes (as is the case for the domains encountered in computational fluid dynamics (CFD)). A regular mesh of such a domain is a mesh where the element sizes in the neighborhood of each component are close to the step-size of this component. As for the element sizes elsewhere in the domain, they must be close to the stepsizes of the discretization of the boundaries situated in some neighborhood.

The second type of discretization concerns the so-called "geometric" discretizations which are adapted to the boundary geometries. In this case, it may be proved that the discretization step-size must be locally proportional to the minimum radius of curvature of the boundary. The drawback of this type of discretization is that a rather wide variation in the discretization step-size may result. In order to avoid this phenomenon, a smoothing technique on the discretization step-size may be applied. For a geometric discretization (not uniform in general) of the domain boundaries, it is tedious to find *a priori* what element sizes make the mesh regular. Obviously, a regular mesh is one where the element sizes are almost constant (or vary just a little). The idea is then to find among all the continuous size functions, that which leads to a minimal variation. This notion of a minimal variation is a characterization, among others, of the surfaces defined by means of harmonic functions.

3.2 From regular mesh to adaptivity

The two above types of meshes appear to be a particular occurrence of a more general mesh generation problem which involves constructing a mesh whose element sizes conform to some given specifications. These requests define, in each point in the domain, the desired element sizes in all directions. There are two types of specifications : *isotropic* and *anisotropic*. In the first case, the size remains constant in all directions (and thus the classical cases fall into this category). As for the second case, the size may vary when the direction varies. This is used for problems where the solution shows large variations in some directions. In both cases, we assume we are given a function $h(P, \overrightarrow{d}) > 0$ defining the size h at point P in the domain following the direction $\overrightarrow{d}$ and the problem comes down to completing a mesh where the edge lengths conform to this function (or size map) h. Written in this way, the problem is not well posed. Indeed, if PQ stands for an edge in the desired mesh, the Euclidean length $||\overrightarrow{PQ}||$ of PQ must satisfy, at the same time, the two antagonist relations :

$$||\overrightarrow{PQ}|| = h(P, \overrightarrow{PQ}) \qquad \text{and} \qquad ||\overrightarrow{PQ}|| = h(Q, \overrightarrow{QP}) \, ,$$

which implies that the vertices P and Q are constructed in such a way as

$$h(P, \overrightarrow{PQ}) = h(Q, \overrightarrow{QP}) \, .$$

This is unlikely to be possible if the size map is such that $\forall P, Q \quad h(P, \overrightarrow{PQ}) \neq h(Q, \overrightarrow{QP})$. In fact, the computation of the Euclidean length of PQ does not take

into account the variation in the map h. Therefore, the mesh construction problem must be formulated in a different way. To this end, we introduce a new definition. A mesh conforms to a size map h if all of its edges have an "average" length equal to the average of the sizes specified along these edges. Thus, it is necessary to define this notion of average length with regard to a size map. To do this, we assume that the data of the map $h(X, \overrightarrow{d})$ $(\forall X, \overrightarrow{d})$ allows for the local definition of a metric tensor (or a metric, in short) $\mathcal{M}(X)$ at X, which in turn defines a new norm which takes into account the size variation related to the directions. Let $l_{\mathcal{M}(X)}(e)$ be the length of edge e computed using metric $\mathcal{M}(X)$. The average length of edge e may be defined as the average of the lengths $l_{\mathcal{M}(X)}(e)$ when X moves along e. If $l_m(e)$ denotes this length (subscript $_m$ for mean), we have :

$$l_m(e) = \frac{|| \int\limits_X l_{\mathcal{M}(X)}(e)\, dX ||}{|| \int\limits_X dX ||} ,$$

for $e = PQ$ and $X = P + t\overrightarrow{PQ}$, we obtain :

$$l_m(PQ) = \int\limits_0^1 l_{\mathcal{M}(P+t\overrightarrow{PQ})}(PQ)\, dt .$$

If $h_m(PQ)$ stands for the average of the sizes along PQ, we have :

$$h_m(PQ) = \int\limits_0^1 h(P + t\overrightarrow{PQ}, \overrightarrow{PQ})\, dt ,$$

and edge PQ conforms to map h if $l_m(PQ) = h_m(PQ)$. To avoid computing $h_m(PQ)$, we redefine metric $\mathcal{M}$ in such a way that $h_m(PQ) = 1$ holds, which implies (in general) $h(X, \overrightarrow{d}) = 1$ $(\forall X, \overrightarrow{d})$ and, in this case, we simply have $l_m(PQ) = 1$. In what follows, we give some remarks about the construction of metric $\mathcal{M}(X)$ starting from the size map $h(X, \overrightarrow{d})$ $(\forall X, \overrightarrow{d})$. The key is to find a metric $\mathcal{M}(X)$ which conforms as well as possible to the map. Let us recall that a metric $\mathcal{M}(X)$ defined at point X is the data of a symmetric positive definite matrix also denoted by $\mathcal{M}(X)$. The geometric locus of the points Y which conform to metric $\mathcal{M}(X)$ at point X is in general an ellipsoïd $\mathcal{E}(X)$ whose equation can be written as :

$${}^t\overrightarrow{XY}\mathcal{M}(X)\overrightarrow{XY} = 1 .$$

This particular expression of the metric prescribes a desired size which is unity in this metric. Indeed, for each point Y in $\mathcal{E}(X)$, we have :

$$l^2_{\mathcal{M}(X)}(XY) = < \overrightarrow{XY}, \mathcal{M}(X)\overrightarrow{XY} > = 1 .$$

The set of points in $\mathcal{E}(X)$ is termed the unit sphere associated with metric $\mathcal{M}(X)$. If $\mathcal{C}(X)$ denotes the geometric locus of the points conforming to the size map $h(X, \overrightarrow{d})$ $(\forall \overrightarrow{d})$ at point X, then metric $\mathcal{M}(X)$ may be characterized as the one whose unit sphere $\mathcal{E}(X)$ has a maximal volume included in $\mathcal{C}(X)$. Such a metric is called the underlying metric. In the particular case where the size map $h(X, \overrightarrow{d})$ only depends on X (isotropic case), metric $\mathcal{M}(X)$ reduces to

$$\mathcal{M}(X) = \frac{1}{h^2(X)}\, \mathcal{I}_d$$

where $\mathcal{I}_d$ is the identity matrix in $\mathbb{R}^d$ ($d = 2$ or 3). In this case, $\mathcal{E}(X)$ and $\mathcal{C}(X)$ are the sphere centered in X whose radius is $h(X)$. When $\mathcal{C}(X)$ is an ellipsoïd (classical anisotropic case), metric $\mathcal{M}(X)$ is obviously the one whose associated sphere $\mathcal{E}(X)$ is similar to $\mathcal{C}(X)$. In the general case where $\mathcal{C}(X)$ is arbitrary, we may make use of optimization algorithms to find $\mathcal{E}(X)$ and thus to have metric $\mathcal{M}(X)$.

The size map $h(X, \overrightarrow{d})$ $(\forall X, \overrightarrow{d})$ is then seen as the metric map $\mathcal{M}(X)$ and a mesh conforming to this metric is a mesh where the edges have an average length unit. A mesh with this property is said to be a *unit mesh*. One could observe that this average edge length in a metric is nothing other than an edge length if we associate a Riemannian structure with the domain, this structure being defined by the metric map $\mathcal{M}(X)$. In this structure, length $L_{\mathcal{M}}$ of edge PQ is given by :

$$L_{\mathcal{M}}(PQ) = \int_0^1 \sqrt{{}^t\overrightarrow{PQ}\,\mathcal{M}(P + t\overrightarrow{PQ})\,\overrightarrow{PQ}}\, dt\ .$$

To summarize, a mesh is said to be conformal to a given size map $h(X, \overrightarrow{d})$ $(\forall X, \overrightarrow{d})$ if it is unity in the Riemannian structure associated with the underlying metric map.

From a practical point of view, the Riemannian structure may be defined in two ways : as a continuous or a discrete structure. The first way consists in defining the metric map $\mathcal{M}$ analytically and, in this case, the metric map $\mathcal{M}$ is explicitly given as a function of the position. The second way consists in defining the map by means of interpolation from the data of the map at the vertices of a mesh, termed the *background mesh*. This approach is popular in adaptive schemes where the metric map is computed in a supporting mesh using an appropriate error estimator. Several interpolation schemes can be used, [11].

To conclude, the question is to know whether a unit mesh conforming to a metric map is suitable for finite element purposes (for instance, in terms of convergence issues). To decide on this point, it is natural to add another criterion to qualify what appears to be a mesh suitable for computation. This criterion is related to the shape of the elements. A unit mesh conforming to a metric map may not be suitable for computational purposes. Indeed, the element shape quality largely depends on the size variation present in the metric map. To avoid this, it is only necessary to modify the metric map, [12], in accordance with the desired size (while preserving certain properties included in the map).

3.3 Unit meshing of a curve or a surface

Throughout this section, we propose a method that results in the construction of a unit mesh in a domain Ω in $\mathbb{R}^d$, $d = 2$ or 3, (the domain being defined by its boundary Σ) equipped with a given Riemannian metric $\mathcal{M}_d$. The method reduces to meshing Ω in such a way that the edges in this mesh are *unity*. Bear in mind that the metric at a point P in Ω is defined by a symmetric positive definite $d \times d$ matrix , $\mathcal{M}_d(P)$. If P is a vertex in the unit mesh of Ω and if PX is an edge with P as an endpoint, then one must have :

$$\int_0^1 \sqrt{{}^t\overrightarrow{PX}\mathcal{M}_d(P + t\overrightarrow{PX})\overrightarrow{PX}}\, dt = 1 .$$

The proposed method involves two steps : the discretization of the boundary Σ of Ω by means of unit elements and the construction of a unit mesh in Ω using as input the above boundary discretization. These two steps are discussed in the following sections. Curve discretization includes two different cases based on the way the curve is defined : in a parametric form or in a discrete form. The same applies for surface discretization.

3.3.1 Unit parametric curve discretization

We assume Σ to be defined by a mathematical (analytical) model. In two dimensions, the boundary Σ of Ω is composed of curved segments $\Gamma_i : I_i \longrightarrow \mathbb{R}^2, t \longmapsto \gamma_i(t)$ where I_i is a closed interval in R and $\gamma_i(t)$ is a continuous function of class C^2. The problem reduces to the discretization of a generic curved segment $\Gamma : I \longrightarrow \mathbb{R}^2, t \longmapsto \gamma(t)$. In three dimensions, the boundary Σ is composed of parametric patches $\Sigma_i : \omega_i \longrightarrow \mathbb{R}^3, (u, v) \longmapsto \sigma_i(u, v)$ where Ω_i is a closed bounded domain in $\mathbb{R}^2$ and $\sigma_i(t)$ is a continuous function of class C^2. Similarly, the problem reduces to the discretization of a generic parametric patch $\Sigma : \omega \longrightarrow \mathbb{R}^3, (u, v) \longmapsto \sigma(u, v)$. In this case, the discretization includes two steps : the discretization of the boundary of Σ which is composed of curved segments in $\mathbb{R}^3$ and that of Σ starting from the discretization of its boundary. Discretizing a curved segment in $\mathbb{R}^2$ is a particular case of the general problem of discretizing a curved segment in $\mathbb{R}^3$. In what follows, we describe how to discretize such segments, then we show that discretizing a parametric patch reduces to constructing a unit mesh of a domain in $\mathbb{R}^2$ in accordance with a metric map induced by the intrinsic properties of the patch.

Discretization of a curved segment in $\mathbb{R}^3$ Let $\Gamma : [a, b] \longrightarrow \mathbb{R}^3, t \longmapsto \gamma(t)$ and let $\gamma(t)$ be a continuous function of class C^2. As previously seen, the length of Γ in the Riemannian structure $\mathcal{M}_3$ is :

$$L(\Gamma) = \int_I \sqrt{{}^t\gamma'(t)\mathcal{M}_3(\gamma(t))\gamma'(t)}\, dt .$$

In order to discretize Γ by means of unit segments, we first compute the integer value n closest to $L(\gamma)$ (thus Γ must be subdivided into n segments), then we compute the values t_i, $1 \leq i \leq n-1$ ($t_0 = a$ and $t_n = b$) such that

$$\frac{L(\Gamma)}{n} = \int_{t_i}^{t_{i+1}} \sqrt{{}^t\gamma'(t)\mathcal{M}_3(\gamma(t))\gamma'(t)}\, dt\,.$$

Finally, the discretization of Γ consists of the straight line segments $\gamma(t_i)\gamma(t_{i+1})$.

3.3.2 Unit discrete curve (re)discretization

There are essentially two ways to define a curve: by a continuous function as explained before or, in a discrete manner, by an ordered set of sampling points. These points can, for instance, be generated by a CAD system or by a scanning device, or they can be the result of a numerical simulation in an adaptive scheme. Generally, the goal is then to obtain from this set of given points a parametric curve which should be as smooth as possible.

In some cases, the data may be "noisy" because of measurement or computation errors, producing an interpolating curve with a rough aspect. Smoothing techniques, including various averaging schemes, can be applied to avoid this phenomenon.

At present, given a set of points $\{P_i\}_{i=0,\ldots,n}$, the problem is to find a continuous function γ defined on R with values in $\mathbb{R}^d$ (in the 2 or 3-dimensional space), and increasing real numbers t_i such that $\gamma(t_i) = P_i$ for each $i = 0, ..., n$. In practice, the solution can be chosen amongst piecewise polynomial functions called splines, by analogy to the draftman's tool consisting of a thin flexible rod made of metal, plastic or wood. Each polynom on an interval $[t_i, t_i + 1]$ is usually of degree 3, being defined by the location of the extremities P_i and P_{i+1}, as are the tangent vectors at these points, thus ensuring a C^1 continuity of γ. To define a tangent vector at each point P_i, several methods are proposed. In the Catmull-Rom approach, the vector is colinear with $\overrightarrow{P_{i-1}P_{i+1}}$ (with particular conditions at both ends P_0 and P_n of the curve). In the de Boor approach, a C^2 continuity is imposed at each intermediate point P_i and a linear system is solved, yielding a smoother aspect to the whole curve. Finally, having a geometric support defined by the interpolating function γ, a new discretization of the curve can be obtained using the method described in the previous section.

3.3.3 Unit parametric surface meshing

Let Σ be a parametric surface defined by: $\sigma : \Delta \longrightarrow \Sigma, (u, v) \longmapsto \sigma(u, v)$, Δ being a domain of $\mathbb{R}^2$, and σ a continuous function of class C^2. We assume that Δ is closed and bounded, as is Σ. The problem we face is to construct a mesh $\mathcal{T}_{\mathcal{M}_3}(\Sigma)$ of Σ that conforms to the metric map $\mathcal{M}_3$. The idea is to construct a mesh in the parametric domain Δ so as to obtain the desired mesh after being mapped onto the surface. To this end, we show that we can define a metric $\widetilde{\mathcal{M}}_2$ in Δ such that

the relation $\mathcal{T}_{\mathcal{M}_3}(\Sigma) = \sigma(\mathcal{T}_{\widetilde{\mathcal{M}}_2}(\Delta))$ is satisfied. To do so, first, we recall the usual Euclidean length formula of a curved segment of $\mathbb{R}^3$ plotted on Σ, then we extend this notion to the case where a Riemannian metric is specified in Σ.

Let Γ be a curved segment of Σ defined by a continuous function of class C^2, $\gamma(t) \in \mathbb{R}^3$, where $t \in [a, b]$. The usual Euclidean length $L_{\mathcal{I}_3}(\Gamma)$ of Γ is given by:

$$L_{\mathcal{I}_3}(\Gamma) = \int_a^b \sqrt{{}^t\gamma'(t)\gamma'(t)}\, dt\,.$$

As Γ is plotted on Σ, there is a function $\omega(t) \in \Omega$, where $t \in [a, b]$, such that $\gamma = \sigma \circ \omega$. We have $\gamma'(t) = \sigma'(\omega(t))\omega'(t)$, where $\sigma'(\omega(t))$ is the 3 by 2 matrix defined as

$$\sigma'(\omega(t)) = \begin{pmatrix} \sigma'_u(\omega(t)) & \sigma'_v(\omega(t)) \end{pmatrix}.$$

Thus, we obtain

$${}^t\gamma'(t)\gamma'(t) = {}^t\omega'(t)\,{}^t\sigma'(\omega(t))\sigma'(\omega(t))\omega'(t)$$

but we have

$${}^t\sigma'(\omega(t))\sigma'(\omega(t)) = \begin{pmatrix} {}^t\sigma'_u(\omega(t)) \\ {}^t\sigma'_v(\omega(t)) \end{pmatrix}\begin{pmatrix} \sigma'_u(\omega(t)) & \sigma'_v(\omega(t)) \end{pmatrix}$$

or

$${}^t\sigma'(\omega(t))\sigma'(\omega(t)) = \mathcal{M}_\sigma(\omega(t))$$

where $\mathcal{M}_\sigma$ is a 2 by 2 matrix which characterizes the local intrinsic metric of Σ at point $\gamma(t)$ and is defined as

$$\mathcal{M}_\sigma(\omega(t)) = \begin{pmatrix} {}^t\sigma'_u(\omega(t))\sigma'_u(\omega(t)) & {}^t\sigma'_u(\omega(t))\sigma'_v(\omega(t)) \\ {}^t\sigma'_v(\omega(t))\sigma'_u(\omega(t)) & {}^t\sigma'_v(\omega(t))\sigma'_v(\omega(t)) \end{pmatrix}.$$

We can deduce that

$$L_{\mathcal{I}_3}(\Gamma) = \int_a^b \sqrt{{}^t\omega'(t)\mathcal{M}_\sigma(\omega(t))\omega'(t)}\, dt\,.$$

The above formula has an interesting interpretation. The Euclidean length of the curved segment $\omega(t)$ plotted in Ω depends on the Euclidean norm of $\omega'(t)$, while the Euclidean length of the curved segment $\gamma(t) = \sigma(\omega(t))$ (image of $\omega(t)$ on Σ) depends on the "Riemannian norm" of $\omega'(t)$ with respect to the local intrinsic metric of Σ.

In particular (depending on the particular reason for generating a mesh), if $\omega(t)$ is a line segment AB of Ω, we have $\omega(t) = A + t\overrightarrow{AB}$, thus $\omega'(t) = \overrightarrow{AB}$, and :

$$L_{\mathcal{I}_3}(\sigma(AB)) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\mathcal{M}_\sigma(A + t\overrightarrow{AB})\overrightarrow{AB}}\, dt\,.$$

The above formula allows us to compute the length of the curved segment on Σ which is the mapping by σ of an edge plotted on Ω. If the new metric $\mathcal{M}_\sigma$ is given in Ω, we have :

$$L_{\mathcal{M}_\sigma}(AB) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\,\mathcal{M}_\sigma(A+t\overrightarrow{AB})\overrightarrow{AB}}\,dt = L_{\mathcal{I}_3}(\sigma(AB))\,.$$

Let us consider the case where a Riemannian metric $\mathcal{M}_3$ of $\mathbb{R}^3$ is specified on Σ. In this case, the Riemannian length $L_{\mathcal{M}_3}(\Gamma)$ of Γ is given by :

$$L_{\mathcal{M}_3}(\Gamma) = \int_a^b \sqrt{{}^t\gamma'(t)\mathcal{M}_3(\gamma(t))\gamma'(t)}\,dt$$

which can be written as :

$$L_{\mathcal{M}_3}(\Gamma) = \int_a^b \sqrt{{}^t\omega'(t)\widetilde{\mathcal{M}}_2(\omega(t))\omega'(t)}\,dt \qquad \text{with} \qquad \widetilde{\mathcal{M}}_2(\omega(t)) = {}^t\sigma'(\omega(t))\mathcal{M}_3(\gamma(t))\sigma'(\omega(t))\,.$$

Thus, if $\omega(t)$ is a line segment AB of Ω, we obtain :

$$L_{\mathcal{M}_3}(\sigma(AB)) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\,\widetilde{\mathcal{M}}_2(A+t\overrightarrow{AB})\overrightarrow{AB}}\,dt\,.$$

The above formula allows us to compute the generalized length of the curved segment on Σ which is the mapping by σ of an edge plotted on Δ. Let us define the new metric $\widetilde{\mathcal{M}}_2$ in Δ. We have

$$L_{\widetilde{\mathcal{M}}_2}(AB) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\,\widetilde{\mathcal{M}}_2(A+t\overrightarrow{AB})\overrightarrow{AB}}\,dt$$

and thus we obtain

$$L_{\mathcal{M}_3}(\sigma(AB)) = L_{\widetilde{\mathcal{M}}_2}(AB)\,.$$

To return to the problem of mesh generation itself, the last equation shows that the mapping of the mesh conforming to the metric $\widetilde{\mathcal{M}}_2$ of Δ onto the surface Σ gives the mesh conforming to the specified metric $\mathcal{M}_3$ of Σ.

3.3.4 Unit discrete surface meshing

The problem of meshing a discrete surface (defined by a triangulation) is closely related to the problem of defining a suitable metric map $\mathcal{M}_3$ on the surface to control mesh modifications. The idea is to extract the intrinsic properties of the underlying surface from the initial surface triangulation.

Formally speaking, the problem is to construct a geometric surface mesh $\mathcal{T}_{\mathcal{M}_3}$ from an initial reference mesh $\mathcal{T}_{ref}$. At first, mesh $\mathcal{T}_{ref}$ is simplified and optimized in accordance with a Hausdorff distance δ, resulting in a geometric reference mesh $\mathcal{T}_{ref,g}$. This first stage aims at removing extra vertices (*i.e.,* those which do not contribute explicitly to the surface definition) and at de-noising the original data. This procedure consists in removing the mesh vertices iteratively, provided two conditions are satisfied : an approximation criterion (related to the hausdorff distance between the current mesh and the original one) and a regularity criterion (related to the local deviation of the mesh edges from the local tangent planes). Then, a geometric support, piecewise C^1 continuous, is defined on mesh $\mathcal{T}_{ref,g}$ so as to define a "smooth" representation of surface Σ. The aim of this support is to supply the location of the closest point onto the surface given a point on a triangular facet. This support will be used to insert a new vertex in the current mesh.

Curvature evaluation. In order to build metric map $\mathcal{M}_3$, we need to evaluate the intrinsic properties of the surface. This requires computing the principal curvatures and principal directions at the vertices of mesh $\mathcal{T}_{ref,g}$. To this end, the surface at a point P is locally approached by a quadric surface, based on a least-square fit of adjacent mesh vertices. The local frame at P is computed based on the normal to a discrete approximation of the normal to the surface at P. To find the coefficients of the quadric, we consider all vertices P_i of the ball of P and assume that the surface fits at best these points. Solving this system is equivalent of minimizing the sum

$$\min \sum_{i}^{m} \left(ax_i^2 + bx_iy_i + cy_i^2 - z_i\right)^2 ,$$

which corresponds to minimizing the square of the norm of the distance to the quadrix surface. Knowing the coefficients a, b and c it becomes easy to find the local curvatures at a point $P = (0, 0, 0)$ in the local frame :

$$\begin{array}{ll} E = 1 + (2au + bv)^2 = 1 , & L = 2a , \\ F = (2au + bv)(bu + 2cv) = 0 , & M = b , \\ G = 1 + (bu + 2cv)^2 = 1 , & N = 2c . \end{array}$$

The analysis of the variations of the normal curvature function

$$\kappa_n(\vec{\tau}) = \frac{\Phi_1(\vec{\tau})}{\Phi_2(\vec{\tau})}$$

leads to resolving a second order equation that admits (in principle) two distinct solutions, 2 pairs (λ_1, κ_1), (λ_2, κ_2). The extrema values κ_1 and κ_2 of κ_n (*i.e.,* the roots of the equation) are the *principal curvatures* of the surface at P. Considering the second order equation

$$\kappa^2 - (\kappa_1 + \kappa_2)\kappa + \kappa_1\kappa_2 = 0$$

yields

$$\begin{aligned} \kappa_1\kappa_2 &= \frac{LN-M^2}{EG-F^2} &&= 4ac-b^2\,, \\ \kappa_1+\kappa_2 &= \frac{NE-2MF+LG}{EG-F^2} &&= 2(a+c)\,, \end{aligned} \tag{1}$$

where $K = \kappa_1\kappa_2$ is the *Gaussian curvature* and $H = \frac{1}{2}(\kappa_1+\kappa_2)$ is the *mean curvature* of the surface at P. Solving these equations allows us to find the extrema values κ_1 and κ_2 at P

$$\kappa_i = \frac{2(a+c) \pm \sqrt{\Delta}}{2}\,,$$

where $\Delta = (2(a+c))^2 - 4(4ac-b^2)$. A similar analysis leads to finding the principal directions at a point P on the surface.

Metric definition. A geometric metric map $\mathcal{M}_3(P)$ can be defined at any mesh vertex P so as to locally bound the gap between the mesh edges and the surface by any given threshold value ε. A matrix of the form

$$\mathcal{M}_3(P)_{\rho_1,\rho_2} = {}^t\mathcal{D}(P) \begin{pmatrix} \frac{1}{\alpha^2\,\rho_1^2(P)} & 0 & 0 \\ 0 & \frac{1}{\beta^2\,\rho_2^2(P)} & 0 \\ 0 & 0 & \lambda \end{pmatrix} \mathcal{D}(P)\,,$$

where $\mathcal{D}(P)$ corresponds to the principal directions at P, $\rho_1 = 1/\kappa_2$, $\rho_2 = 1/\kappa_2$ are the main radii of curvature, α and β are appropriate coefficients and $\lambda \in \mathbb{R}$, provides an *anisotropic* (curvature-based) control of the geometry. This discrete metric prescribes mesh sizes as well as element stretching directions at mesh vertices. The local size is proportional to the principal radii of curvature, the coefficient of proportionality being related to the largest allowable deviation gap between the mesh elements and the surface geometry [16]. For instance, setting a constant gap values comes down to fixing

$$\alpha = 2\sqrt{\varepsilon(2-\varepsilon)} \quad \text{together with} \quad \beta = 2\sqrt{\varepsilon\frac{\rho_1}{\rho_2}\left(2-\varepsilon\frac{\rho_1}{\rho_2}\right)}\,.$$

As the size may change rapidly from one vertex to another, the mesh gradation may not be bounded locally. To overcome this problem, size map $\mathcal{M}_3$ is modified using a size-correction procedure [12].

Surface remeshing. Having defined an adequate metric $\mathcal{M}_3$ at any mesh vertex, the discrete surface meshing problem consists in constructing a unit mesh with respect to metric map $\mathcal{M}_3$. To this end, local mesh modifications are applied, based on the edge length analysis. The optimization consists in collapsing the short edges and splitting the large edges based on their relative length. Geometric measures are

used to control the deviation between the mesh elements and the surface geometry as well as the element shape quality [16].

3.4 Unit volume meshing

The global scheme for unit mesh generation is well-known : a coarse mesh (without internal points) of the domain is constructed using a classical Delaunay method, and then this mesh is enriched by adding the field points before being optimized. The field points are defined in an iterative manner. At each iteration step, these points are created using a method for edge saturation or an advancing-front method. Then, they are inserted in the current mesh using the *constrained Delaunay kernel*, [23] in a Riemannian context. This process is repeated as long as the current mesh is being modified.

3.4.1 Point placement strategies

Here we mention two methods that allow the field points to be defined. The advantage of the first is its simplicity and low cost. On the other hand, the second method generally results in better quality meshes.

Edge saturation At each iteration step, the field points are defined in such a way as to subdivide the edges in the current mesh by unit length segments. A field point is retained if it is not too close (*i.e.,* at a distance less than 1) to a point already existing in the mesh or previously retained for insertion.

Advancing-front strategy At each iteration step, a set of facets (edges in two dimensions) in the current mesh, thus a front, is retained and the field points are created from this front so as to form unit elements (elements with unit edges). A facet in the current mesh is considered as a front entity if it separates a unit element from a non-unit element. A face front being identified, a classical advancing-front strategy is used to create a field point.

3.4.2 Point insertion

In a classical problem, the constrained Delaunay kernel based on cavity re-meshing is written as $\mathcal{T} = \mathcal{T} - \mathcal{C}(P) + \mathcal{B}(P)$ where $\mathcal{C}(P)$ is the cavity associated with point P and $\mathcal{B}(P)$ is the re-meshing of $\mathcal{C}(P)$ based on P ($\mathcal{T}$ being the current mesh).

An extension of this approach involves redefining cavity $\mathcal{C}(P)$ in a Riemannian context [23]. To this end, we modify the previously introduced *Delaunay measure* α by $\alpha_{\mathcal{M}_d}$ associated with pair (P, K), with respect to a metric $\mathcal{M}_d$, which now reads :

$$\alpha_{\mathcal{M}_d}(P, K) = \left[\frac{d(O_K, P)}{r_K} \right]_{\mathcal{M}_d},$$

where O_K (resp. r_K) is the center (resp. radius) of the circumsphere of K and $[*]_{\mathcal{M}_d}$ indicates that the quantity $*$ is measured in the Euclidean space characterized by metric $\mathcal{M}_d$. Then, cavity $\mathcal{C}(P)$ is defined as it was in the classical case. Note that the correction algorithm already mentioned, [20], is then mandatory.

3.4.3 Optimization processes

The proposed method results in a unit mesh of domain Ω. Nevertheless, the mesh quality can be improved by means of two optimization processes, one made up of topological modifications, the other consisting of geometrical modifications. The first mainly involves applying a number of facet flips while the other consists of node repositioning.

Edge length quality. Let AB be a mesh edge. The length quality Q_l of AB in the Riemannian metric $\mathcal{M}_d$ may be defined as :

$$Q_l(AB) = \begin{cases} L_{\mathcal{M}_d}(AB) & \text{if } L_{\mathcal{M}_d}(AB) \leq 1 \\ \dfrac{1}{L_{\mathcal{M}_d}(AB)} & \text{if } L_{\mathcal{M}_d}(AB) > 1 \end{cases}$$

With this measure, $0 \leq Q_l(AB) \leq 1$ holds and a unit edge has a length quality with a value of 1. This quality measure about the edge lengths shows how the mesh conforms to the specified Riemannian metric $\mathcal{M}_d$.

The edge length quality of a mesh $\mathcal{T}$ is defined by :

$$Q_l(\mathcal{T}) = (\frac{1}{|\mathcal{T}|} \sum_{e \in \mathcal{T}} Q_l(e) \, , \min_{e \in \mathcal{T}} Q_l(e))$$

where e stands for an edge in mesh $\mathcal{T}$ and $|\mathcal{T}|$ is the number of such edges. The two quantities in the formula measure respectively the average and the minimum of the length qualities of the mesh edges.

Element shape quality. Let K be a mesh element. In the classical Euclidean space, a popular measure for the shape quality of K is :

$$Q_f(K) = c \, \frac{V(K)}{\sum\limits_{e(K)} l^2(e(K))} \, ,$$

where $V(K)$ denotes the volume of K, $e(K)$ being the edges in K and c is a scaling coefficient such that the quality of a regular element has the value of 1. With this definition, we have $0 \leq Q_f(K) \leq 1$ and a nicely shaped element has a quality close to 1 while a ill-shaped element has a quality close to 0.

In a Riemannian space, the quality of element K can be defined by

$$Q_f(K) = \min_{1 \leq i \leq d+1} Q_f^i(K) \, ,$$

where $Q_f^i(K)$ is the element quality in the Euclidean space associated with the metric $\mathcal{M}_d^i$ corresponding to the vertex number i in K.

To measure quality $Q_f^i(K)$, it is simply necessary to transform the Euclidean space related to the metric specified at vertex i of K into the usual Euclidean space and to consider the quality value of element K^i associated with K, in other words :

$$Q_f^i(K) = Q_f(K^i) .$$

It is easy to show that :

$$Q_f^i(K) = c\,\frac{\sqrt{Det(\mathcal{M}_d^i)}\,V(K)}{\sum\limits_{e(K)} l^2_{\mathcal{M}_d^i}(e(K))} ,$$

Similarly, the shape quality of the elements in mesh $\mathcal{T}$ is defined by :

$$Q_f(\mathcal{T}) = (\frac{1}{|\mathcal{T}|}\sum_{K\in\mathcal{T}} Q_f(K)\ ,\ \min_{K\in\mathcal{T}} Q_f(K))$$

where K stands for an element in mesh $\mathcal{T}$. The two quantities in the formula measure respectively the average and the minimum shape qualities of the mesh elements.

Facet flip. Facet flip only affects the mesh topology. It has proved to be very efficient for shape quality improvement. This technique results in the removal of a facet of arbitrary dimensionality when this is possible. Let f be a facet of arbitrary dimensionality in the mesh. We use the term *shell* (cf. [23]) for f, the set of elements sharing f. Flipping f involves constructing a triangulation of the hull of the shell of f where f is not a mesh entity. The quality of a shell is that of its worst element. The flip is then processed if the quality of the new triangulation is better than that of the initial shell.

When a Riemannian metric must be followed, it is necessary to sort these facet flips while this is not strictly necessary in a classical Euclidean case. This leads to associating the expected ratio of improvement β_f with face f by emulating a flip. Then, to optimize the mesh, an iterative process is used which applies the flips in the decreasing order of the above ratios. To begin with, the ratio of improvement is set at a given value $\omega > 1$, then ω is modified and decreases to 1. Such a strategy leads to flipping in first the most significative operations in terms of mesh improvement.

Point repositioning. Let P be an internal point in the mesh and let (K_i) be the set of elements with P as a vertex (*i.e.,* the *ball* associated with P, cf. [23]). Repositioning P consists in moving P so as to enhance the quality of the worst elements in (K_i). Two methods can be advocated for node repositioning. One based on unit length, the other on optimal elements. The first method improves

the edge length quality for the elements in (K_i) while the second improves the shape of these elements. In practice, both methods are applied to all the internal points of the mesh.

Let (P_i) the set of vertices in (K_i) other than P. With each point P_i is associated the optimal point (P_i^*) such that

$$\overrightarrow{P_iP_i^*} = \frac{1}{L_{\mathcal{M}_d}(P_iP)}\overrightarrow{P_iP},$$

for which $L_{\mathcal{M}_d}(P_iP_i^*) = 1$ holds. Repositioning P consists of moving point P step by step (*i.e.,* using a relaxation scheme) towards the centroid of the points (P_i^*) if the quality of the worst element in (K_i) is improved. This process results in unit edge lengths for the edges that have P as one endpoint.

Let (f_i) be the facets in the elements in (K_i) which are opposite vertex P. *i.e.,* $(K_i = [P, f_i])$. With each facetf_i, is associated the optimal point P_i^* such that element $K_i^* = [P_i^*, f_i]$ satisfies

$$Q_f(K_i^*) = \max_{\tilde{P}} Q_f([\tilde{P}, f_i]),$$

where $\tilde{P}$ is an arbitrary point located on the same side of f_i as P is. Similarly, repositioning P consists of moving P step by step towards the centroid of points (P_i^*) if the quality of the worst element in (K_i) is improved. This process results in optimal quality elements for the elements in the ball of P.

To find point P_i^* we can consider the centroid of the optimal points related to the f_is, each of which is evaluated in the Euclidean structure related to the metric defined at a vertex of K_i.

Conclusion. A quality mesh is defined with respect to the metric in hand and, therefore, the notion of a quality depends on the targeted application. In this sense, other quality measures are meaningless (such as angles, Delaunay criterion, aspect ratio, ...) when they are considered alone.

4 Adaptive FEM computations

Let us consider a bounded domain Ω described by its boundary surface Γ. In the context of a numerical simulation performed on domain Ω using a mesh of Ω as spatial support, we suggest a general scheme including an adaptive meshing loop of Ω. This scheme is made up of two distinct parts. The first part only involves the generation of an initial mesh of the computational domain Ω. The second part concerns an adaptation loop including the computation, the *a posteriori* error estimation and the generation of adapted meshes. The method advocated in what follows is a $h-$method where the parameter of adaptation is h, e.g. the size (or the directional sizes) of the mesh elements.

4.1 Initial mesh of Ω

The problem consists of constructing a mesh of Ω from an initial reference mesh $\mathcal{T}_{ref}(\Gamma)$ of its boundary Γ and from a metric map $\mathcal{M}_0(\Gamma)$ indicating the desired element sizes on the surface. To construct this initial mesh $\mathcal{T}_0(\Omega)$, we proceed in several steps :

- the initial reference mesh $\mathcal{T}_{ref}(\Gamma)$ is simplified and optimized in a given Hausdorff envelope, so as to obtain a geometric reference mesh $\mathcal{T}_{ref,g}(\Gamma)$ of Γ.
- a geometric "smooth" support (piecewise G^1 continuous) is defined on mesh $\mathcal{T}_{ref,g}(\Gamma)$, so as to obtain a smooth geometric representation of boundary Γ;
- metric map $\mathcal{M}_0(\Gamma)$, supplied on mesh $\mathcal{T}_{ref,g}(\Gamma)$ is then rectified so as to be compatible with the surface geometry;
- the rectified map $\mathcal{M}_0(\Gamma)$ is again modified to account for the desired mesh gradation;
- mesh $\mathcal{T}_{ref,g}(\Gamma)$ is adapted in terms of element sizes to the modified map $\mathcal{M}_0(\Gamma)$ so as to obtain the initial computational mesh $\mathcal{T}_0(\Gamma)$;
- volume mesh $\mathcal{T}_0(\Omega)$ is generated from mesh $\mathcal{T}_0(\Gamma)$ associated with the metric map $\mathcal{M}_0(\Gamma)$.

This schematic flow is illustrated in Figure 1, where one can see the data flowchart related to the input and the output of the various procedures involved in the entire process.

4.2 General diagram of an adaptive computation

The adaptation loop aims to capture and to refine the physical solution of the numerical simulation performed on domain Ω. In general, a computation performed on mesh $\mathcal{T}_0(\Omega)$ does not allow a satisfactory solution to be obtained. Hence, we suggest the following iterative adaptation scheme, in which, at each iteration step i :

- a computation is performed on mesh $\mathcal{T}_i(\Omega)$ leading to the solutions field $\mathcal{S}_i(\Omega)$;
- solution $\mathcal{S}_i(\Omega)$ is analyzed using an adequate error estimator and a metric map $\mathcal{M}_i(\Omega)$ is deduced prescribing the element sizes in order to obtain a subsequent solution at a given accuracy;
- the metric map $\mathcal{M}_i(\Gamma)$ restricted to surface Γ is rectified with regard to the respect of the surface geometry;
- the (partially) rectified metric map $\mathcal{M}_i(\Omega)$ is modified to take into account the specified mesh gradation;
- surface mesh $\mathcal{T}_i(\Gamma)$ is adapted, in terms of element sizes, to the metric modified map $\mathcal{M}_i(\Gamma)$ to obtain mesh $\mathcal{T}_{i+1}(\Gamma)$;

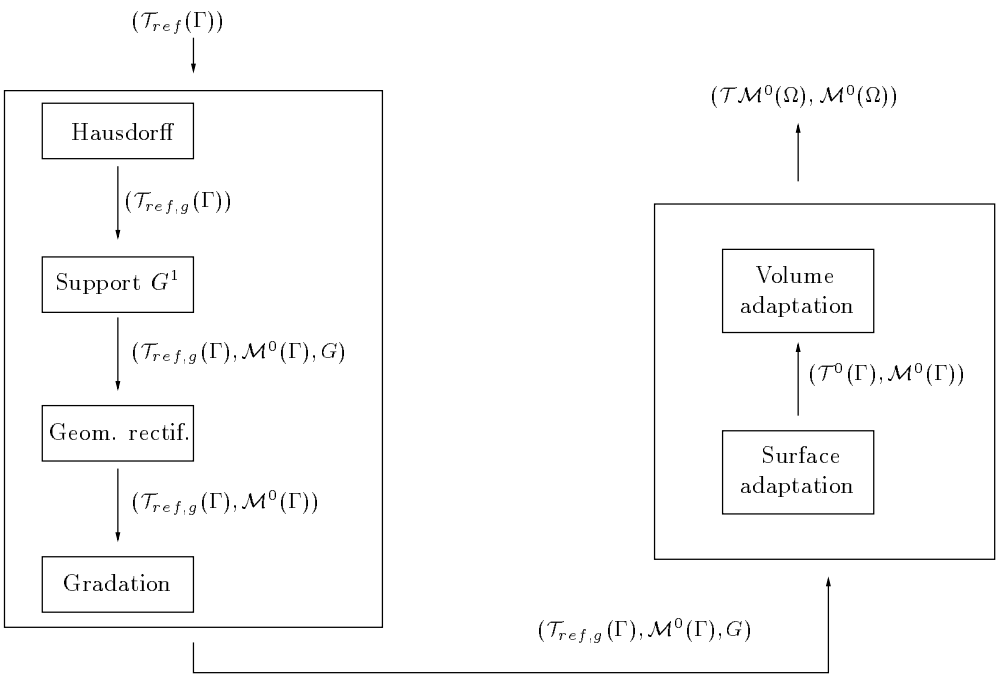

Figure 1: *The various stages of the construction of the initial (volume) mesh of Ω : from the initial mesh $\mathcal{T}_{ref}(\Gamma)$ of the boundary to the computational mesh $\mathcal{T}_0(\Omega)$ and the associated metric map $\mathcal{M}_0(\Omega)$.*

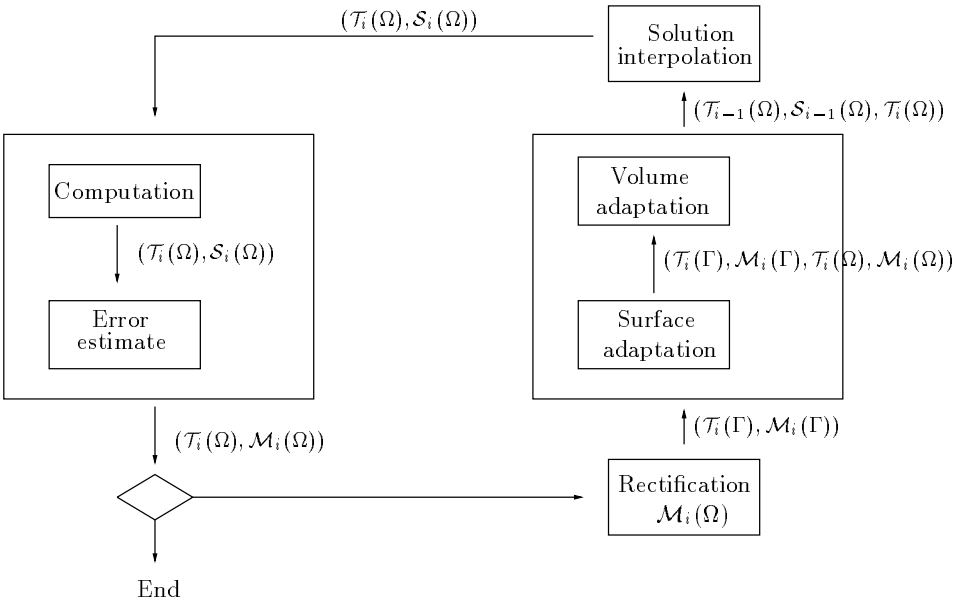

Figure 2: *The various stages of the general adaptation scheme.*

- volume mesh $\mathcal{T}_{i+1}(\Omega)$ adapted to the metric field $\mathcal{M}_i(\Omega)$ is generated from mesh $\mathcal{T}_i(\Gamma)$ associated with its metric map $\mathcal{M}_i(\Gamma)$;
- solution $\mathcal{S}_i(\Omega)$ associated with mesh $\mathcal{T}_i(\Omega)$ is interpolated on mesh $\mathcal{T}_{i+1}(\Omega)$.

These various stages are illustrated in Figure 2.

4.3 Error estimates

There exist several error estimate strategies which make it possible to control *a posteriori* the error with the solution computed on a finite element mesh, [15]. These estimators can be used to control the mesh by means of a *h-adaption* method in such a way that the resulting solution is of a given accuracy.

Among these estimators are those based on the interpolation error (and, therefore, of a purely geometric nature as the operator itself is not considered). This class of estimators has been studied by various authors, [5], [4], [34], [7]. Nevertheless, most of these papers require a parametre, h, the element size, to be small or to vanish and therefore are asymptotic results. The estimator then makes use of appropriate Taylor expansions and provides information about the desired size, h. However, as this size is not necessarily small we propose a new approach which, while close, does not necessitate any peculiar assumption about this parameter and, therefore, is likely to be more justified. Our approach is to some extent close to solutions used in a different topic, the construction of meshes in parameterized patches, see [35], [2], [10], among others.

4.3.1 Problem statement and state of the art

Let Ω be a domain in R^d (where $d = 1$, 2 or 3) and let $\mathcal{T}$ be a simplicial mesh of Ω where the simplices are linear, P^1, or quadratic, P^2, elements. We assume we have in hand the solution denoted by u (to meet the classical notations in error estimate literature) of a finite element computation previously done in Ω using $\mathcal{T}$ as a mesh, this solution being scalar values denoted by $u_{\mathcal{T}}$. Let u be the exact solution, the problem first involves computing the gap $e_{\mathcal{T}} = u - u_{\mathcal{T}}$ from u to $u_{\mathcal{T}}$ which represents the underlying error due to the finite element approximation. Then, it involves constructing a new mesh $\mathcal{T}'$ such that the estimated gap from u to solution $u_{\mathcal{T}'}$, as obtained using a new computation, is bounded by a given threshold. Specific issues include :

- how to evaluate gap $e_{\mathcal{T}}$ from u and $u_{\mathcal{T}}$
- how to use this information to construct a new mesh such that the corresponding gap is bounded in the desierd range.

The finite element solution $u_{\mathcal{T}}$ is not interpolant (e.g. the solution at the nodes of $\mathcal{T}$ is not coincident with the exact value u at these nodes). Moreover, for any element in the mesh it is not possible to guarantee that $u_{\mathcal{T}}$ is coincident with the exact value of u at at least one point in this element. It is therefore tedious to explicitly access the gap of $e_{\mathcal{T}}$. However, the direct analysis of this gap has been

studied in a number of works, [39]. But, in general, it remains an open problem. Therefore, other non direct approaches have been proposed to quantify or bound this gap. Let $\widetilde{u}_{\mathcal{T}}$ be the interpolate of u over mesh $\mathcal{T}$ (here is a linear or a quadratic piecewise function after the degree of the elements in $\mathcal{T}$) and let $\widetilde{e}_{\mathcal{T}}$ be the gap $u-\widetilde{u}_{\mathcal{T}}$ from u to $\widetilde{u}_{\mathcal{T}}$, the so-called *interpolation error* about u for $\mathcal{T}$. To obtain the range of gap $e_{\mathcal{T}}$, we assume the following relation to be satisfied :

$$||e_{\mathcal{T}}|| \leq C\,||\widetilde{e}_{\mathcal{T}}||\,,$$

where $||.||$ stands for a norm and C is a constant independent of $\mathcal{T}$. In other words, we assume the finite element error to be majored by the interpolation error. This allows us to simplify the initial problem by considering the following problem : given $\widetilde{u}_{\mathcal{T}}$ the interpolation of u over mesh $\mathcal{T}$, how can we construct another mesh $\mathcal{T}'$ where the interpolation error is bounded by a given threshold ? As $\widetilde{u}_{\mathcal{T}}$ can be seen as a discrete representation of u, the problem reduces to find a characterization of the meshes where this interpolation error is bounded. This topic has been addressed in various papers, such as [7], and, in most of them, using a "measure" of the interpolation error makes it possible to find some constraints to which the mesh elements must conform. In the context of mesh adaptation methods, we are mainly interested in h-methods or adaptation in size where these constraints are expressed in terms of element sizes. Some classical measures of this error, classified into two categories, continuous or discrete, together with the corresponding constraints about the mesh elements are discussed in [7]. It turns out that the discrete approach is more convenient from the point of view of mesh adaptation.

In this chapter, we first propose a majoration of the interpolation error in two dimensions which extends to arbitrary dimensions (and is close to a work by Anglada *et al.*, [2]). Then, we show how this majoration can be used to adapt the mesh. After which, we introduce a new measure for quantifying the interpolation error which depends on the local deformation of the Cartesian surface associated with the solution. We demonstrate how this measure makes it possible to control the interpolation error in H^1 norm and, therefore, is likely to be more appropriate.

4.3.2 A bound in two dimensions

We consider a mesh made up of piecewise linear triangles. Let K be a mesh element and let u be the mapping from R^2 to R, assumed to be sufficiently smooth. We note by $\Pi_h u$ the linear interpolant of u over K and we assume u and $\Pi_h u$ to be coincident at the vertices of K. The aim is to bound $|(u-\Pi_h u)(x)|$ for x in K.

An isotropic bound. In various references and, for instance, in [22], it is shown that

$$e = ||(u-\Pi_h u)(x)||_\infty \leq \frac{2}{9}\,L^2\,M\,, \tag{2}$$

where L is the longest edge in K and M is given by (H_u being the Hessian of u)

$$M = \max_{x \in K} \left(\max_{||\overrightarrow{v}||=1} |\langle \overrightarrow{v} \, , \, H_u(x) \overrightarrow{v} \rangle| \right) .$$

After (2), a gap in the range of ε for e implies that L is such that

$$L^2 \leq \frac{9\,\varepsilon}{2\,M} \, , \tag{3}$$

while observing that if M is zero, any value of L is convenient. Therefore, after the value of h_{max}, the diameter of element K, as compared with L, one can know whether triangle K is suitable, too small or too large. Therefore, a size adaptation can be envisaged if necessary.

However, in practice, the difficulty is to evaluate M using $\Pi_h u$ by means of :

- approaching $\nabla_u(a)$, $\nabla_u(b)$ and $\nabla_u(c)$,
- approaching $H_u(a)$, $H_u(b)$ and $H_u(c)$,
- approaching M by the largest eigenvalue of matrices $|H_u(a)|$, $|H_u(b)|$ and $|H_u(c)|$ where $|H_u|$ is constructed from H_u after being made positive definite.

Similarly, vectors $\nabla_u(a)$, $\nabla_u(b)$ and $\nabla_u(c)$ and matrices $H_u(a)$, $H_u(b)$ and $H_u(c)$ can be approximated using generalized finite differences (a variation of the Green formula, [?]) of function $\Pi_h u$. However, the Taylor expansion about a, b and c can be used. To this end, we assume $K = [a, b, c]$ to be an element inside the mesh and we denote by (a_i) the set of the mesh vertices adjacent to a (thus including b and c). Writing the Taylor expansion with respect to a in each a_i, yields the following over-determined system

$$\left\{ \; u(a_i) \; \approx \; u(a) + \langle \overrightarrow{aa_i} \, , \, \nabla_u(a) \rangle \right.$$

which is equivalent to the minimization problem

$$\min_{\overrightarrow{z}} \sum_i \, w_i^2 \, (\langle \overrightarrow{aa_i} \, , \, \overrightarrow{z} \rangle - u(a_i) + u(a))^2$$

where w_i is a weight measuring the influence of equation i in the system. The solution of this problem is that of the linear system ${}^t A \, W \, A \, \overrightarrow{z} \; = \; W \, \beta$ where A is the matrix made up of the row vectors $(\overrightarrow{aa_i})$, W being the diagonal matrix related to the weights (w_i^2) and β being the vector with components $(u(a_i) - u(a))$. Similarly, vector $\nabla_u(a)$ being known, Hessian $H_u(a)$ can be computed by the linear system

$$\left\{ \; u(a_i) \; \approx \; u(a) + \langle \overrightarrow{aa_i} \, , \, \nabla_u(a) \rangle \; + \; \tfrac{1}{2} \langle \overrightarrow{a\,a_i} \, , \, H_u(a) \, \overrightarrow{aa_i} \rangle \, . \right.$$

which is equivalent to a similar minimization problem. Vectors $\nabla_u(b)$ and $\nabla_u(c)$ as well as matrices $H_u(b)$ and $H_u(c)$ are obtained in the same way.

Due to the isotropic nature of this result, using this kind of estimator may lead to a un-necesarily fine mesh. In fact, M is the largest eigenvalue of the family of

operators $|H_u|$ (in all points in K) and therefore L is the size corresponding to this value. As a consequence, an anisotropic phenomemon will be considered in an isotropic way while imposing, in all directions, a size equal to the smallest size related to the eigenvalues.

An anisotropic bound. We assume that vertex a of K is the *site* of x (*i.e.*, x is closer to a than to b and c) to be the point where a maximal gap occurs. We also assume x to be in K (and not in one edge of K, which leads to a similar result). Then we note a' the point intersection of the line supporting ax with the edge opposite a, e.g. edge bc in K. We expand e in a from x by means of the Taylor expansion with integral :

$$e(a) = (u - \Pi_h u)(a) = (u - \Pi_h u)(x) + \langle \overrightarrow{xa} \, , \, \nabla_u (u - \Pi_h u)(x) \rangle$$

$$+ \int_0^1 (1-t) \, \langle \overrightarrow{ax} \, , \, H_u(x + t\overrightarrow{xa}) \, \overrightarrow{ax} \rangle \, dt \, .$$

As a is the site of x, the scalar value λ such that $\overrightarrow{ax} = \lambda \overrightarrow{aa'}$ is smaller than $\dfrac{2}{3}$, therefore

$$|e(x)| \, = \, | \int_0^1 (1-t) \, \lambda^2 \, \langle \overrightarrow{aa'} \, , \, H_u(a+t\overrightarrow{xa}) \, \overrightarrow{aa'} \rangle \, dt \, | \, \leq \frac{4}{9} | \int_0^1 (1-t) \, \langle \overrightarrow{aa'} \, , \, H_u(a+t\overrightarrow{xa}) \, \overrightarrow{aa'} \rangle \, dt \, |$$

which yields

$$|e(x)| \leq \frac{2}{9} \, \max_{y \in K} | \, \langle \overrightarrow{aa'} \, , \, H_u(y) \, \overrightarrow{aa'} \rangle \, | \, , \tag{4}$$

After (4), imposing a gap of ε for e, leads for triangle $K = [a, b, c]$

$$\max_{y \in K} | \, \langle \overrightarrow{aa'} \, , \, H_u(y) \, \overrightarrow{aa'} \rangle \, | \, \leq \frac{9}{2} \varepsilon \, ,$$

where a' is defined as above. This inequality is satisfied if

$$\max_{y \in K} \langle \overrightarrow{aa'} \, , \, |H_u(y)| \, \overrightarrow{aa'} \rangle \, \leq \frac{9}{2} \varepsilon \, .$$

Let $\mathcal{M}(a)$ be the symmetric positive definite matrix such that

$$\max_{y \in K} \langle \overrightarrow{az} \, , \, |H_u(y)| \, \overrightarrow{az} \rangle \, \leq \langle \overrightarrow{az} \, , \, \mathcal{M}(a) \, \overrightarrow{az} \rangle$$

for all z in K and such that the region (e.g. bounded by the corresponding ellipse) defined by

$$\langle \overrightarrow{az} \, , \, \mathcal{M}(a) \, \overrightarrow{az} \rangle \, \leq \frac{9}{2} \varepsilon$$

has a minimal surface. Then $\mathcal{M}(a)$ results in a size constraint or again a metric in a which varies after the directions.

In these equations, we have assumed the site of point x where gap e is maximal to be a. As x is not known, the sites b and c must be taken into account in turn. This leads to the system

$$\left\{\begin{array}{l} \max\limits_{y\in K} \langle \overrightarrow{aa'}\,,\, |H_u(y)|\,\overrightarrow{aa'}\rangle \;\leq\; \frac{9}{2}\varepsilon \\ \max\limits_{y\in K} \langle \overrightarrow{bb'}\,,\, |H_u(y)|\,\overrightarrow{bb'}\rangle \;\leq\; \frac{9}{2}\varepsilon \\ \max\limits_{y\in K} \langle \overrightarrow{cc'}\,,\, |H_u(y)|\,\overrightarrow{cc'}\rangle \;\leq\; \frac{9}{2}\varepsilon \end{array}\right.$$

where b' and c' are points respectively in edges ac and ab. As above, we can define the metrics $\mathcal{M}(a)$, $\mathcal{M}(b)$ and $\mathcal{M}(c)$ in a, b and respectively c such that :

$$\left\{\begin{array}{l} \max\limits_{y\in K} \langle \overrightarrow{aa'}\,,\, |H_u(y)|\,\overrightarrow{aa'}\rangle \;\leq\; \langle \overrightarrow{aa'}\,,\, \mathcal{M}(a)\,\overrightarrow{aa'}\rangle \;\leq\; \frac{9}{2}\varepsilon \\ \max\limits_{y\in K} \langle \overrightarrow{bb'}\,,\, |H_u(y)|\,\overrightarrow{bb'}\rangle \;\leq\; \langle \overrightarrow{bb'}\,,\, \mathcal{M}(b)\,\overrightarrow{bb'}\rangle \;\leq\; \frac{9}{2}\varepsilon \\ \max\limits_{y\in K} \langle \overrightarrow{cc'}\,,\, |H_u(y)|\,\overrightarrow{cc'}\rangle \;\leq\; \langle \overrightarrow{cc'}\,,\, \mathcal{M}(c)\,\overrightarrow{cc'}\rangle \;\leq\; \frac{9}{2}\varepsilon \end{array}\right.$$

Therefore, for triangle K, a metric $\mathcal{M}(K)$ can be constructed in such a way as, on the one hand, equations

$$\left\{\begin{array}{l} \langle \overrightarrow{v}\,,\, \mathcal{M}(K)\,\overrightarrow{v}\rangle \;\leq\; \langle \overrightarrow{v}\,,\, \mathcal{M}(a)\,\overrightarrow{v}\rangle \\ \langle \overrightarrow{v}\,,\, \mathcal{M}(K)\,\overrightarrow{v}\rangle \;\leq\; \langle \overrightarrow{v}\,,\, \mathcal{M}(b)\,\overrightarrow{v}\rangle \\ \langle \overrightarrow{v}\,,\, \mathcal{M}(K)\,\overrightarrow{v}\rangle \;\leq\; \langle \overrightarrow{v}\,,\, \mathcal{M}(c)\,\overrightarrow{v}\rangle \end{array}\right.$$

are satisfied for any vector $\overrightarrow{v}$ and, on the other hand, the surface $\langle \overrightarrow{v}\,,\, \mathcal{M}(K)\,\overrightarrow{v}\rangle = 1$ is maximal. In other words, the metric in K is the largest size constraint along all the directions satisfying the size constraints at the vertices a, b and c.

Actually, for the sake of simplicity, we consider $\mathcal{M}(a) = |H_u(a)|$, $\mathcal{M}(b) = |H_u(b)|$ and $\mathcal{M}(c) = |H_u(c)|$, matrix H_u being determined as in the isotropic case. Metric $\mathcal{M}(K)$ can be defined as the intersection of the three metrics $\mathcal{M}(x)$ for $x = a, b$ and c (cf. [17]).

Remark. In three dimensions, the constant $\frac{2}{9}$ should be $\frac{9}{32}$.

4.3.3 A surface based approach

In the previous sections we proposed a majoration about the interpolation error based on the Hessian of the solution, which has been directly used to obtain the size constraints about the mesh elements. In the present section, we propose a new approach by considering an appropriate Cartesian surface.

Let Ω be the computational domain, let $\mathcal{T}(\Omega)$ be a mesh of Ω and let $u(\Omega)$ be the physical solution obtained in Ω via mesh $\mathcal{T}(\Omega)$. The pair $(\mathcal{T}(\Omega), u(\Omega))$ allows us to define a Cartesian surface $\Sigma_u(\mathcal{T})$ (we assume u to be a scalar function). Given $\Sigma_u(\mathcal{T})$, the problem of minimizing the interpolation error consists in defining an (optimal) mesh $\mathcal{T}_{opt}(\Omega)$ in Ω such that surface $\Sigma_u(\mathcal{T}_{opt})$ is as smooth as possible.

To this end, we propose a local characterization of the surface near a vertex. Two methods are introduced, the first using the local deformation allows for an isotropic adaptation while the other, using the local curvature, results in an anisotropic adaptation.

Local deformation of a surface. The basic idea consists in a local characterization of the deviation (at order 0) of surface mesh $\Sigma_u(\mathcal{T})$ near a vertex with respect to a reference plane, in specific the tangent plane of the surface at this vertex. This deviation can be evaluated by considering the Hessian along the normal to the surface (e.g. the second fundamental form).

Let P be a vertex in the solution surface $\Sigma_u(\mathcal{T})$. Locally, near P, this surface has a parameterized form $\sigma(x,y)$, (x,y) being the parameters, with $P = \sigma(0,0)$. Using a Taylor expansion at order 2 of σ near P, results in

$$\sigma(x,y) \;=\; \sigma(0,0) \;+\; \sigma'_x\, x \;+\; \sigma'_y\, y \;+\; \frac{1}{2}\,(\sigma''_{xx}\, x^2 \;+\; 2\,\sigma''_{xy}\, x\, y \;+\; \sigma''_{yy}\, y^2) \;+\; o(x^2+y^2)\, e$$

where $e = (1,1,1)$. If $\nu(P)$ stands for the normal to the surface at P, then quantity $\langle \nu(P), (\sigma(x,y) - \sigma(0,0))\rangle$ represents the gap from point $\sigma(x,y)$ to the tangent plane in P and cn be written as

$$\frac{1}{2}\,(\langle \nu(P), \sigma''_{xx}\rangle\, x^2 \;+\; 2\,\langle \nu(P), \sigma''_{xy}\rangle\, x\, y \;+\; \langle \nu(P), \sigma''_{yy}\rangle\, y^2) \;+\; o(x^2+y^2)$$

which is therefore proportional to the second fundamental form of the surface when x^2+y^2 is small enough.

The local deformation of the surface at P is defined as the maximal gap of the vertices adjacent to P to the tangent plane of the surface at P. If (P_i) denotes those vertices, then the local deformation $\varepsilon(P)$ of the surface at P is given by :

$$\varepsilon(P) \;=\; \max_i \langle \nu(P), \overrightarrow{PP_i}\rangle\,.$$

Therefore the optimal mesh of $\Sigma_u(\mathcal{T})$ for Ω is a mesh where the size at all nodes p is inversely proportional to $\varepsilon(P)$ where $P = (p, u(p))$. Formally speaking, the optimal size $h_{opt}(p)$ associated with node p is written

$$h_{opt} \;=\; h(p)\,\frac{\epsilon}{\varepsilon(P)}$$

where ε is the given threshold and $h(p)$ is the size of the elements near p in mesh $\mathcal{T}(\Omega)$.

As can be seen, the local deformation is a rather easy way to characterize the local deviation of the surface which does not involve computing the explicit computation of the Hessian of the solution. The only drawback in this measure is that it allows only an isotropic adaptation. In the same context (minimizing the local deviation), using the curvature allows an analysis of this deviation which is both more precise and anisotropic.

Local curvature of a surface. Analyzing the local curvature of the surface related to the solution also makes it possible to minimize the deviation (order 1) from the tangent planes of the solution which interpolate the exact solution. Indeed, while considering the construction of isotropic surface meshes, we have shown in [10] how these two deviations, of order 0 and 1, are bounded by a given threshold and how the element size at all vertices is proportional to the minimal radius of curvature. Let $P = (p, u(p))$ be a vertex in $\Sigma_u(\mathcal{T})$, let $\rho_1(P)$ and $\rho_2(P)$ with $\rho_1(P) \leq \rho_2(P)$, be the two principal radii of curvature and let $(\overrightarrow{e_1}(P), \overrightarrow{e_2}(P))$ the corresponding unit principal directions. The ideal size at P is

$$h^{\Sigma}_{opt}(P) \;=\; \gamma\, \rho_1(P)$$

where γ is a factor related to the specified threshold about the deviation. This size is defined in the tangent plane at the surface at P. Let us consider the frame $(P, \overrightarrow{e_1}(P), \overrightarrow{e_2}(P))$ in this plane, if $h^{\Sigma}_{opt}(P)$ reads $h^{\Sigma}_{opt}(P) = h^{\Sigma}_1 \overrightarrow{e_1}(P) + h^{\Sigma}_2 \overrightarrow{e_2}(P)$ in this frame then the constraint in size at P can be written in $(P, \overrightarrow{e_1}(P), \overrightarrow{e_2}(P))$ by :

$$\left(\begin{array}{cc} h^{\Sigma}_1 & h^{\Sigma}_2 \end{array} \right) \frac{\mathcal{I}_2}{\gamma^2 \rho_1^2(P)} \left(\begin{array}{c} h^{\Sigma}_1 \\ h^{\Sigma}_2 \end{array} \right) \;=\; 1$$

which is the equation of a circle centered at P in the tangent plane at the surface at P. By means of an orthogonal projection of this circle in the plane of Ω, we obtain the size constraint at p. If $\overrightarrow{v_1}(p)$ and $\overrightarrow{v_2}(p)$ are the orthogonal projections of $\overrightarrow{e_1}(P)$ and $\overrightarrow{e_2}(P)$ in the plane of Ω, then this size constraint in frame $(p, \overrightarrow{i}, \overrightarrow{j})$ ($\overrightarrow{i} = (1,0)$ et $\overrightarrow{j} = (0,1)$) is given by

$$\left(\begin{array}{cc} h_1 & h_2 \end{array} \right)^t \left(\left(\begin{array}{cc} \overrightarrow{v_1}(p) & \overrightarrow{v_2}(p) \end{array} \right)^{-1} \right)^t \frac{\mathcal{I}_2}{\gamma^2 \rho_1^2(P)} \left(\begin{array}{cc} \overrightarrow{v_1}(p) & \overrightarrow{v_2}(p) \end{array} \right)^{-1} \left(\begin{array}{c} h_1 \\ h_2 \end{array} \right) \;=\; 1$$

where (h_1, h_2) are the coordinates in the frame $(p, \overrightarrow{i}, \overrightarrow{j})$ of the projection of h^{Σ}_{opt} in the plane of Ω. This relationship defines a metric (which is in general anisotropic) at p.

The metric previously defined may lead to a large number of elements due to the isotropic nature of the elements in the surface. In order to minimize this number of elements, and for anisotropic meshing purposes, a similar relationship involving the two principal radii of curvature can be exhibited, [17]. In such a case, the ideal size of the surface element is given using a so-called geometric metric which, at vertex P of $\Sigma_u(\mathcal{T})$, is written

$$\left(\begin{array}{cc} h^{\Sigma}_1 & h^{\Sigma}_2 \end{array} \right) \left(\begin{array}{cc} \dfrac{1}{\gamma^2 \rho_1^2(P)} & 0 \\ 0 & \dfrac{1}{\eta^2(\gamma, \rho_1(P), \rho_2(P)) \rho_2^2(P)} \end{array} \right) \left(\begin{array}{c} h^{\Sigma}_1 \\ h^{\Sigma}_2 \end{array} \right) \;=\; 1\,.$$

In this expression, γ is a factor related to the given threshold about the deviation and $\eta(\gamma, \rho_1(P), \rho_2(P))$ is a function related to γ, $\rho_1(P)$ and $\rho_2(P)$ which ensures a similar deviation along the two principal directions. This relationship generally

describes an ellipse in the tangent plane of the surface at P which includes the circle obtained in the isotropic case. Similarly, the corresponding metric at p is obtained after a projection of this ellipse in the plane of Ω.

In practice, computing the local curvature at all vertices in this surface first leads to computing the normal (then the gradient) by means of a weighted average of the unit normal at the adjacent elements. Then, in the local frame (the tangent plane together with the normal) we construct a quadratic centered at this vertex which passes at best through the adjacent vertices. After which we consider locally the Hessian to be that of this quadric. Finally, using the gradient and the Hessian at the nodes of $\mathcal{T}(\Omega)$, we compute the principal curvatures and directions at all vertices in surface $\Sigma_u(\mathcal{T})$.

4.4 Solution interpolation

In the proposed adaptive schema, it is necessary to interpolate the current solution from the former mesh to the new one in order to continue the computation at a given stage.

This step involves interpolating solution $\mathcal{S}_i(\Omega)$ associated with mesh $\mathcal{T}_i(\Omega)$ on $\mathcal{T}_{i+1}(\Omega)$. In the case where there is no physical constraints in the PDE problem under solution, the interpolation problem is written as a simple optimization problem like

$$\min \|\mathcal{S}_{i+1}(\Omega) - \mathcal{S}_i(\Omega)\|,$$

where $\|.\|$ is a norm, for instance, a L_2 or a H_1 Sobolev norm, and each solution $\mathcal{S}$ is associated with its corresponding mesh. The solution of this minimization problem necessitates computing the intersection of mesh $\mathcal{T}_i(\Omega)$ and mesh $\mathcal{T}_{i+1}(\Omega)$. In cases where the physical constraints must be considered then the underlying problem is a constrained minimization problem. This is well understood for linear physical constrained operators, [6].

However, in practice, a simple linear interpolation of $\mathcal{S}_i(\Omega)$ on $\mathcal{T}_{i+1}(\Omega)$ allows a solution which is close to the ideal targeted solution. This linear interpolation does not require complex computations such as explicitly computing the mesh intersections.

5 Application examples

A number of application examples are given to demonstrate the approaches we have proposed. Figure 3 illustrates two stages of a mesh for a transonic calculation (turbulent Navier-Stokes) in two dimensions while Figure 4 considers an example in three dimensions where isotropic adaptation is employed to solve a steady compressible Euler problem. Figure 5 considers the same case but, in this case, an anisotropic adaptation scheme has been used. It could be interesting to compare these two adaptive computations in terms of number of elements and CPU requirements for the Euler solver (using a Work-station), the CPU for the mesh generation

steps being widely insignifiant. In this respect, the isotropic case used a series of meshes ranging from $200,000$ to $900,000$ elements for a CPU time of about 7 hours while the anisotropic case necessitated a series of meshes ranging from $35,000$ to $90,000$ elements for a CPU time of about 25 minutes. One could notice that doing the same calculation without adaptation may need using very large meshes and parallel processing or, even, may be impossible.

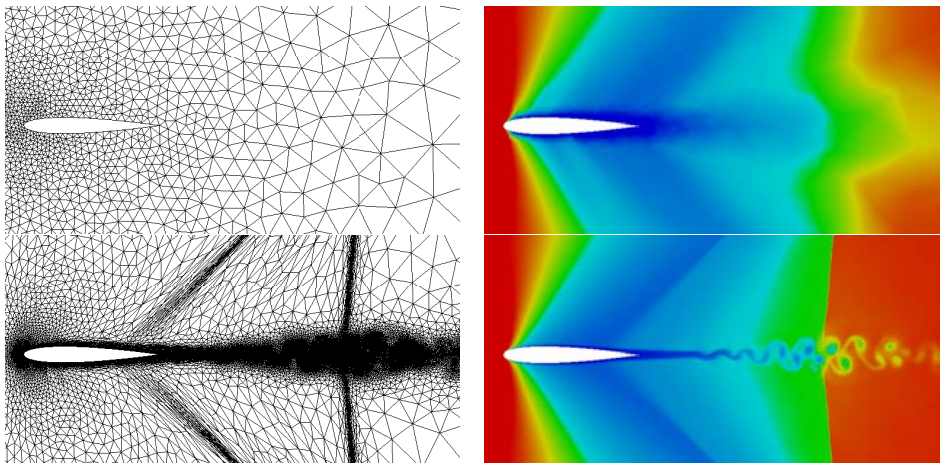

Figure 3: *Transonic flow around a Naca0012 wing, (initial and adapted) meshes and iso-densities.*

6 Conclusions

In this paper we have discussed Delaunay based mesh generation methods and mesh adaptivity issues for automated planar, surface and volume meshing. After a review of the classical mesh generation methods we have considered adaptive schemes where the solution is to be accurately captured.

Application examples have been shown to demonstrate the various approaches proposed throughout the paper.

References

[1] F. Alauzet, P.L. George, B. Mohammadi, P.J. Frey and H. Borouchaki, Transient fixed point-based unstructured mesh adaptation, *Int. J. Numer. Meth. Fluids*, **43**, 729-745, 2003.

[2] M.V. Anglada, N.P. Garcia and P.B. Crosa, Directional adaptive surface triangulation, *Computer Aided Geometric Design*, **16**, 107-126, 1999.

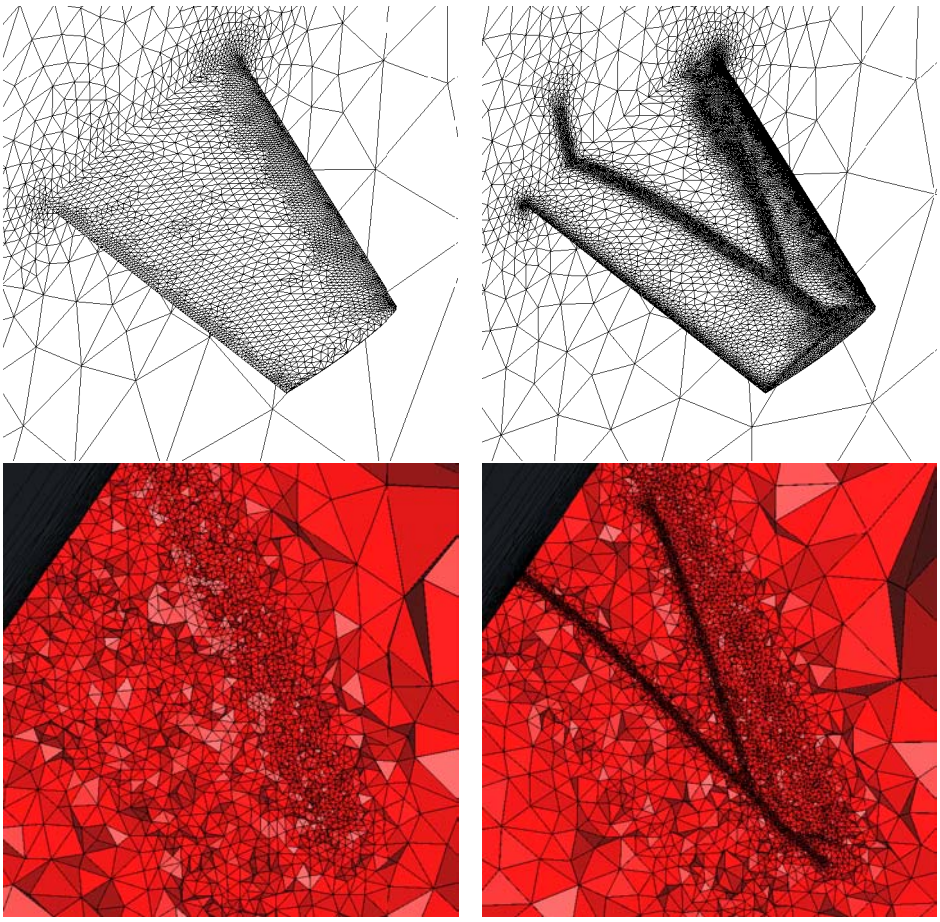

Figure 4: *Transonic flow around a ONERA M6 wing in three dimensions by means of* **isotropic** *adaptation. Surface meshes and cuts through the tet mesh at iterations #1 and #9.*

[3] T. Apel, *Anisotropic Finite Element : Local Estimates and Applications*, Wiley Teubner, 1999.

[4] E.F. D'Azevedo and B. Simpson, On optimal triangular meshes for minimizing the gradient error, *Numerische Mathematik*, **59**(4), 321-348, 1991.

[5] I. Babuska and A. Aziz, On the angle condition in the finite element method, *SIAM J. Numer. Analysis*, **13**, 214-227, 1976.

[6] R.E. Bank, Mesh smoothing using *a posteriori* estimates, *Siam J. numer. anal.*, **34**(3), 979-997, 1997.

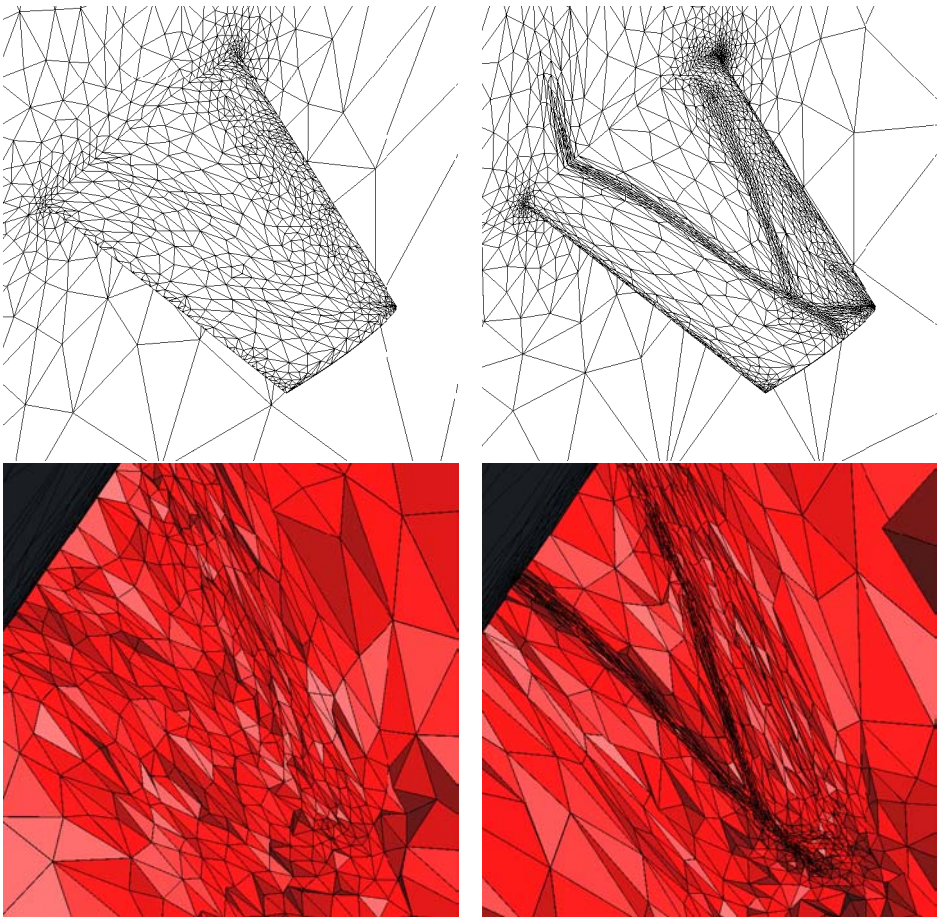

Figure 5: *Transonic flow around a ONERA M6 wing in three dimensions by means of* **anisotropic** *adaptation. Surface meshes and cuts through the tet mesh at iterations #1 and #9.*

[7] M. Berzins, Mesh quality : a function of geometry, error estimates or both ?, *Engineering with Computers*, **15**, 236-247, 1999.

[8] J.D. Boissonnat and M. Yvinec, *Algorithmic Geometry*, Cambridge University Press, 1997.

[9] H. Borouchaki and P.L. George, Quality mesh generation, *C.R. Acad. Sci. Paris*, Concise review paper, t. 328, Serie II-b, pp. 505-518, 2000.

[10] H. Borouchaki, D. Chapelle, P.L. George, P. Laug et P. Frey, Estimateur d'erreur géométrique et adaptation, in *Maillage et adaptation*, Traité Mécanique et Ingénierie des Matériaux, Hermès-Lavoisier, in french. Paris, 2001.

[11] H. Borouchaki, P.L. George, F. Hecht, P. Laug and E. Saltel, Delaunay mesh generation governed by metric specifications. Part I. Algorithms, *Finite Elements in Analysis and Design*, **25**, 61-83, 1997.

[12] H. Borouchaki, F. Hecht and P.J. Frey, Mesh Gradation Control, *Int. J. Numer. Meth. Eng.*, **43**, 1143-1165, 1997.

[13] G.F. Carey, *Computational grids : generation, adaptation and solution strategies*, *Taylor and Francis*, 1997.

[14] P.G. Ciarlet, Basic Error Estimates for Elliptic Problems, in Handbook of Numerical Analysis, vol II, P.G. Ciarlet and J.L. Lions Eds, North Holland, 17-352, 1991.

[15] M. Fortin, Estimation a posteriori et adaptation de maillages, *Revue européenne des éléments finis*, **9**(4), 2000.

[16] P.J. Frey and H. Borouchaki, Geometric surface mesh optimization, *Computing and Visualization in Science*, **1**, 113-121, 1998.

[17] P.J. Frey and P.L. George, Mesh generation, *Hermès, France*, also in french. 2000.

[18] P.L. George, *Automatic mesh generation. Applications to finite element methods*, Wiley, 1991.

[19] P.L. George, Sur une bascule tridimensionnelle anisotrope et ses applications, *RT. Inria* , n^o 273, 2002.

[20] P.L. George and F. Hermeline, Delaunay's mesh of a convex polyhedron in dimension d. Application to arbitrary polyhedra, *Int. j. numer. methods eng.*, **33**, 975-995, 1992.

[21] P.L. George, Automatic Mesh Generation and Finite Element Computation, in Handbook of Numerical Analysis, vol IV, Finite Element methods (Part 2), Numerical Methods for Solids (Part 2), P.G. Ciarlet and J.L. Lions Eds, North Holland, pp. 69-190, 1996.

[22] P.L. George (Eds.), *Maillage et adaptation*, Traité Mécanique et Ingénierie des Matériaux, Hermès-Lavoisier, in french. Paris, 2001.

[23] P.L. George and H. Borouchaki, Delaunay Triangulation and Meshing, Application to Finite Element, *Hermès, France*, also in french. 1998.

[24] P.L. George, H. Borouchaki and E. Saltel, "Ultimate" robustness in meshing an arbitrary polyhedron, *Int. j. numer. methods eng.*, **58**, 1061-1089, 2003.

[25] F. Hermeline, Une méthode automatique de maillage en dimension n, Thèse Paris VI, 1980.

[26] B. Joe, Construction of three-dimensionnal Delaunay triangulations using local transformations, *Comput. Aided Geom. Design*, **8**, 123-142, 1991.

[27] S.H. Lo, A new mesh generation scheme for arbitrary planar domains, *Int. j. numer. methods eng.*, **21**, 1403-1426, 1985.

[28] R. Löhner and P. Parikh, Three-Dimensional Grid Generation by the Advancing Front Method, *Int. j. numer. methods fluids*, **8**, 1135-1149, 1988.

[29] R. Löhner, Automatic Unstructured Grid Generators, *Finite Elements in Analysis and Design*, **25**(3-4), 111-134, 1997.

[30] D.L. Marcum and N.P. Weatherill, Unstructured grid generation using iterative point insertion and local reconnection, *AIAA Journal.*, **33**(9), 1619-1625, 1995.

[31] J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz, Adaptive remeshing for compressible flow computations, *Jour. of Comput. Phys.*, **72**, 449-466, 1987.

[32] J. Peraire, J. Peiro, K. Morgan, Adaptive remeshing for three-dimensional compressible flow computations, *Jour. of Comput. Phys.*, **103**, 269-285, 1992.

[33] F.P. Preparata and M.I. Shamos, Computational geometry, an introduction, Springer-Verlag, 1985.

[34] S. Rippa, Long and thin triangles can be good for linear interpolation, *SIAM J. Numer. Analysis*, **29**, 257-270, 1992.

[35] X. Sheng and B.E. Hirsch , Triangulation of trimmed surfaces in parametric space, *Comput. Aided Des.*, **24**(8), 437-444, 1992.

[36] J.F. Thompson, B.K. Soni and N.P. Weatherill, *Handbook of grid generation*, CRC Press, 1999.

[37] M.G. Vallet, Génération de maillages éléments finis anisotropes et adaptatifs. *Thèse Université Paris VI*, 1992.

[38] N. Van Phai, Automatic mesh generation with tetrahedron elements, *Int. J. Numer. Meth. Eng.*, **18**, 237-289, 1982.

[39] R. Verfürth, *A review of a posteriori error estimation and adaptive refinement techniques*, Wiley Teubner, 1996.

[40] D.F. Watson, Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes, *Comput. J.*, **24**, 167-172, 1981.

[41] N.P. Weatherill and O. Hassan, Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints, *Int. j. numer. methods eng.*, **37**, 2005-2039, 1994.

[42] M.A. Yerry and M.S. Shephard, A modified quadtree approach to finite element mesh generation, *IEEE Computer Graphics Appl.*, **3**(1), 39-46, 1983.

[43] M.A. Yerry and M.S. Shephard, Automatic three-dimensional mesh generation by the modified-octree technique, *Int. j. numer. meth. eng.*, **20**, 1965-1990, 1984.

[44] O.C. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, London, 1977.